

50



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/045,188	10/18/2001	Chia-Hsin Li	AP110HO	4860

20178 7590 08/03/2005

EPSON RESEARCH AND DEVELOPMENT INC
INTELLECTUAL PROPERTY DEPT
150 RIVER OAKS PARKWAY, SUITE 225
SAN JOSE, CA 95134

EXAMINER

LESNIEWSKI, VICTOR D

ART UNIT	PAPER NUMBER
----------	--------------

2152

DATE MAILED: 08/03/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/045,188

Applicant(s)

LI ET AL.

Examiner

Victor Lesniewski

Art Unit

2152

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 07 June 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-27 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-27 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

1. The amendment filed 6/7/2005 has been placed of record in the file.
2. Claims 9 and 11 have been amended.
3. Claims 25-27 have been added.
4. Claims 1-27 are now pending.
5. The applicant's arguments with respect to claims 1-24 have been fully considered but they are not persuasive. A detailed discussion is set forth below.

Response to Amendment

6. Claim 9 has been amended to show that the method passes parameters to a port and has a program obtain the parameters by listening to the port. The amendment proves a change in scope to the independent claim as the independent claim now explicitly states having the program obtain the parameters by listening to the port. However, similar limitations have been previously presented in other claims and none of the amended claims show a patentable distinction over the prior art of record.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2152

8. Claims 1-6, 9-13, 15, 17-23, and 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spyker et al. (U.S. Patent Number 6,571,389), hereinafter referred to as Spyker, in view of Giroir et al. (U.S. Patent Number 6,854,006), hereinafter referred to as Giroir.

9. Spyker disclosed a method for improving the manageability and usability of Java environments. In an analogous art, Giroir disclosed a system for allowing target applications to be locally selected and easily accessed. Both systems deal with the downloading of Java-based applications.

10. Concerning claims 1, 9, 15, 20, and 25, Spyker did not explicitly disclose determining if the latest version of an application is present on a client. Although Spyker states the ability to retrieve a latest version of the properties for an application (see column 4, lines 64-67), he is not explicit about checking the version of the application. However, a major focus of Giroir's system is to check for the latest version on the client. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to modify the system of Spyker by adding the ability to determine if the latest version of an application is present on a client as provided by Giroir. Here the combination satisfies the need for a Java environment that is easier to use. See Spyker, column 4, lines 3-10. This rationale also applies to those dependent claims utilizing the same combination.

11. Some claims will be discussed together. Those claims which are essentially the same except that they set forth the claimed invention as a computer readable media are rejected under the same rationale applied to the described claim.

12. Thereby, the combination of Spyker and Giroir discloses:

- <Claims 1 and 20>

A method for installing and launching a network application, through a distributed network, the application contained on a server, the method comprising: accessing the server through the network (Spyker, column 11, lines 43-47); selecting parameters of the application (Spyker, column 11, lines 54-58); determining if the latest version of the application is present on a client (Giroir, column 9, lines 17-28); downloading and installing an archive file containing the latest version of the application if the latest version of the application is not present on the client (Spyker, column 12, lines 30-50); launching the application (Spyker, column 14, lines 42-44); and passing parameters to a port, the port in communication with the application, wherein the application is configured to access native libraries of the archive file (Spyker, column 15, lines 12-20).

- <Claims 2 and 21>

The method as recited in claim 1, wherein passing parameters to a port further includes: configuring the application to listen to the port; and sending the parameters over the port, the parameters being sent by a control module (Spyker, column 7, line 53 through column 8, line 3).

- <Claims 3 and 22>

The method as recited in claim 2, wherein the port is a TCP/IP port (Spyker, column 7, line 66 through column 8, line 3).

- <Claims 4 and 23>

The method as recited in claim 1, wherein the application is a Java application, the Java application configured to be executed by a Java virtual machine of an operating system of the client (Spyker, column 1, lines 22-44).

- <Claim 5>

The method as recited in claim 1, wherein the archive file is a .cab file, the .cab file including the application and a control module (Spyker, column 3, line 65 through column 4, line 2).

- <Claim 6>

The method as recited in claim 5, wherein the application is a Java application and the .cab file includes native libraries, the native libraries configured to standardize an output of the Java application across platforms (Spyker, column 15, lines 12-20 where the use of the JAR type is akin to the use of a CAB type file).

- <Claim 9>

A method for running a network program on a client, the network program accessed through a web browser, the method comprising: accessing a server via a port, said server containing the network program (Spyker, column 11, lines 43-47); configuring parameters of the application (Spyker, column 11, lines 54-58); linking to a page on the server, the page containing version information of a server archive file (Spyker, column 10, lines 13-24 and column 11, line 56 through column 12, line 29); determining if a client archive file is present on the client, the determining further including; inspecting the client archive file if the client archive file is present to ascertain if the client archive

file is the same version as the server archive file (Giroir, column 9, lines 17-28); and downloading the server archive file to the client if the client archive file is not present or if the client archive file is not the same version as the server archive file (Spyker, column 12, lines 30-50); launching the program (Spyker, column 14, lines 42-44); passing the parameters to the port (Spyker, column 15, lines 12-20); and having the program obtain the parameters by listening to the port (Spyker, column 7, line 53 through column 8, line 3).

- <Claim 10>

The method as recited in claim 9, wherein the network program is a Java based program for a printing application (Giroir, column 7, lines 59-62).

- <Claim 11>

The method as recited in claim 9 further including: executing the application, the executing further including, accessing native libraries, the native libraries installed by the archive file (Spyker, column 15, lines 12-20).

- <Claim 12>

The method as recited in claim 11, wherein the port is a TCP/IP port and the application is configured to listen to the TCP/IP port so that the application can receive the parameters passed to the port (Spyker, column 7, line 53 through column 8, line 3).

- <Claim 13>

The method as recited in claim 9, wherein the archive file is a .cab file, the .cab file containing a control module, the control module configured to pass the parameters to a port (Spyker, column 3, line 65 through column 4, line 2).

- <Claim 15>

A system for installing and launching an application through a network, the system comprising: a server (Spyker, figure 2, item 47), the server including an application contained in a server archive file (Spyker, column 8, lines 57-59), the application including a plurality of options (Spyker, column 9, lines 34-52), the server configured to link to a page containing version information of the server archive file (Spyker, column 9, lines 23-33 and column 10, lines 13-24), the server further configured to allow a user to select the options of the application (Spyker, column 11, lines 42-43); a web browser (Spyker, column 1, lines 45-48), and a client (Spyker, figure 2, item 10), the client in communication with the server through the web browser (Spyker, figure 2, item 50), the client inspected by the server to determine if a client archive file is present and current by comparing the version information of the server archive file with version information of the client archive file (Giroir, column 9, lines 17-28 and Spyker, column 11, line 56 through column 12, line 29), wherein if the client archive file is not present or not current, the server archive file is downloaded to the client, the client archive file including the application (Spyker, column 12, lines 30-50), the application further configured to listen to a port such that any of the options selected by a user are transmitted to the application by a control module through the port (Spyker, column 7, line 53 through column 8, line 3).

- <Claim 17>

The system as recited in claim 15, wherein the application is a Java application containing printing functionality (Giroir, column 7, lines 59-62).

- <Claim 18>

The system as recited in claim 17, wherein the Java application uses a Java virtual machine of an operating system of the client (Spyker, column 1, lines 22-44).

- <Claim 19>

The system as recited in claim 17, wherein the printing functionality is regulated by native libraries, the native libraries included in the client archive file (Spyker, column 15, lines 12-20).

- <Claim 25>

A method for installing and launching a network application, through a distributed network, the application contained on a server, the method comprising: using a client computing device to establish a communication link through a network port to the server on the network (Spyker, column 11, lines 43-47); selecting parameters of the application (Spyker, column 11, lines 54-58); determining if the latest version of the application is present on a client (Giroir, column 9, lines 17-28); downloading and installing an archive file containing the latest version of the application if the latest version of the application is not present on the client (Spyker, column 12, lines 30-50); launching the application (Spyker, column 14, lines 42-44); passing the parameters to the network port (Spyker, column 15, lines 12-20 and column 11, line 62 through column 12, line 1); having the application monitor the network port for said parameters, and implement the parameters (Spyker, column 7, line 53 through column 8, line 3 and column 11, line 62 through column 12, line 1).

Art Unit: 2152

- <Claim 26>

The method of claim 25, wherein: the client computing device uses an web browser to establish said communication link with the server (Spyker, column 11, lines 43-47); and said application is made independent of said web browser (Spyker, column 8, lines 10-40).

- <Claim 27>

The method of claim 26, wherein said application is a stand-alone Java application having access to native program libraries unimpaired by web browser limitations (Spyker, column 8, lines 10-27 and column 15, lines 12-20).

Since the combination of Spyker and Giroir discloses all of the above limitations, claims 1-6, 9-13, 15, 17-23, and 25-27 are rejected.

13. Claims 7, 8, 14, 16, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spyker in view of Giroir, as applied above, further in view of Schmidt et al. (U.S. Patent Number 6,535,894), hereinafter referred to as Schmidt.

14. The combination of Spyker and Giroir disclosed a technique where Java programs can be executed without relying on the use of a browser to provide a run-time environment, in which the client can be checked for the latest version before installation. In an analogous art, Schmidt disclosed a method for updating archive files. Both methods deal primarily with archive files in a Java-based networking environment.

15. Concerning claims 7 and 24, the combination of Spyker and Giroir did not explicitly disclose the link containing version information as an HTML page. However, the use of HTML

in such a system was well known in the art as evidenced by Schmidt who states the use of HTML files in conjunction with his system for updating archived files. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to modify the combination of Spyker and Giroir by adding the ability to link to an HTML page as provided by Schmidt. Again, the combination satisfies the need for a Java environment that is easier to use. See Spyker, column 4, lines 3-10. This rationale also applies to those dependent claims utilizing the same combination.

16. Concerning claim 14, the combination of Spyker and Giroir did not explicitly disclose the use of a digital signature. However, Schmidt disclosed an archived file that can contain digital signatures. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to modify the combination of Spyker and Giroir by adding the ability to use digital signatures as provided by Schmidt. Again, the combination satisfies the need for a Java environment that is easier to use. See Spyker, column 4, lines 3-10.

17. Concerning claim 16, the combination of Spyker and Giroir did not explicitly disclose the use of INF files within the archived files. However, Schmidt's JAR file, related to a CAB file, contains a sub-directory for INF information that aids in the installation and launching of the application. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to modify the combination of Spyker and Giroir by adding the ability to use INF files as provided by Schmidt. Again, the combination satisfies the need for a Java environment that is easier to use. See Spyker, column 4, lines 3-10.

18. Thereby, the combination of Spyker, Giroir, and Schmidt discloses:

- <Claims 7 and 24>

The method as recited in claim 1, wherein selecting parameters of the application further includes: linking to an HTML page the HTML page including an object containing version information of an archive file (Spyker, column 9, lines 23-33 and column 10, lines 13-24 and Schmidt, column 7, lines 13-19).

- <Claim 8>

The method as recited in claim 7, wherein the object includes the width and height of a window of a web browser where the application appears (Schmidt, column 7, lines 59-63).

- <Claim 14>

The method as recited in claim 13, wherein the control module is further configured to be digitally signed (Schmidt, column 9, lines 3-5).

- <Claim 16>

The system as recited in claim 15, wherein the client and server archive files are .cab files, the .cab files including .INF files, the .INF files including a launcher application containing the control module (Spyker, column 3, line 65 through column 4, line 2 and Schmidt, column 8, lines 53-67).

Since the combination of Spyker, Giroir, and Schmidt discloses all of the above limitations, claims 7, 8, 14, 16, and 24 are rejected.

Response to Arguments

19. In the remarks, the applicant has argued:

- <Argument 1>

Spyker does not disclose the features of claim 1 because he does not disclose “passing parameters to a port, the port in communication with the application, wherein the application is configured to access native libraries of the archive file” as recited in claim 1.

- <Argument 2>

Spyker does not disclose the features of claim 15 because he does not disclose “the application further configured to listen to a port such that any of the options selected by a user are transmitted to the application by a control module through the port” as recited in claim 15.

- <Argument 3>

Spyker does not disclose the features of claim 2 because he does not disclose “configuring the application to listen to the port; and sending the parameters over the port, the parameters being sent by a control module” as recited in claim 2.

- <Argument 4>

Spyker does not disclose the features of claim 4 because he does not disclose “the Java application configured to be executed by a Java virtual machine of an operating system of the client” as recited in claim 4.

20. In response to argument 1, Spyker does disclose the features as recited in claim 1. As per the discussion on page 10 of the remarks, it is noted that the use of JNI alone in Spyker is not being directly equated to the ports as claimed. The previous line citation, column 15, lines 12-20, shows Spyker’s ability to allow for Java applications to call native libraries. Thus the use of

JNI in the system satisfies the limitation “wherein the application is configured to access native libraries of the archive file.” This citation further states the passing of environment variables and application parameters which, in Spyker’s system, do get passed through ports in communication with an application, thus satisfying the port limitations of the claim. This citation should be taken into consideration with the other citations listed for the claim as the other citations for the preceding limitations describe further support for the port functionality. For example, column 11, lines 42-66 states additional detail on the passing of parameters or dependencies between the server and client. Spyker passes parameters using TCP/IP which utilizes specific ports in communication with the application.

21. Furthermore, the applicant has stated that Spyker describes “the use of Java applets, not stand-alone applications” on page 12 of the remarks. This statement is incorrect. Although Spyker’s Jobbi programs may be related to applets in that they utilize advantages of both applets and applications, they are not the same as normal Java applets. In fact, a major difference is that they are stand-alone programs as Spyker clearly states that the Jobbi programs do not use a browser as an execution environment. See column 8, lines 6-18. Thus the run-time environment of each Jobbi program can operate independently of a web browser.

22. In response to argument 2, Spyker does disclose the features as recited in claim 15. The previous line citation, column 7, line 53 through column 8, line 3, states the use of dynamic loading software which controls data transfer (or parameter passing) between the client and server. Again, it is noted that Spyker utilizes TCP/IP which controls port communications and directs communications to or from the appropriate application. Furthermore, TCP/IP operates in such a way that one side listens to the port for incoming data appropriate for the corresponding

Art Unit: 2152

application. Thus it can be seen that the software modules operating in Spyker's system along with his use of TCP/IP disclose the limitations of listening to a port and transmitting selected options to the application through the port as recited in claim 15.

23. In response to argument 3, Spyker does disclose the features as recited in claim 2.

Similarly to the discussion of argument 2, it can be seen that Spyker's system utilizes TCP/IP and thus the system maintains both the ability to listen to the port and the ability to pass parameters over the port as claimed. The applicant has stated that Spyker describes the use of JNI which does not use ports. However, Spyker's use of HTTP running on TCP/IP to transfer information between the client and server is different from his use of JNI. The applicant has also pointed out that Spyker states that his system "operates independently of the communications protocol used to send messages or files between the client and server." The applicant believes that "this is in direct conflict with the present invention, which requires that the Java application listen to the port (which is part of the communication protocol used to send messages between a client and a server) to obtain its operating parameters." Here the applicant has misinterpreted the reference. When Spyker states that his system operates independently of the communications protocol, he means that the appropriate Jobbi program modules operate in the same way despite the communications protocol used. Thus Spyker says that his system is made to operate with any of various communications protocols. He is not implying that his system does not use a communications protocol. The exemplary embodiment of Spyker's disclosure in fact utilizes TCP/IP. Here the use of TCP/IP is not "in direct conflict" with the applicant's invention, but rather meets the limitations as claimed concerning the use of ports.

24. The same reasoning is applied to claim 3 where the applicant has again stated that Spyker uses JNI, not ports. Spyker in fact uses TCP/IP ports as discussed in the preceding paragraphs.

25. In response to argument 4, Spyker does disclose the features as recited in claim 4. Although the previous line citation, column 1, lines 22-44, appears in the background of Spyker's invention, the discussion of Java virtual machines is relevant to the disclosed system. Spyker's system allows for multiple run-time environments that exist on a client to execute his Jobbi programs. The Jobbi program meets the limitation of a Java application in the claim, while the run-time environment meets the limitation of a JVM. For clarification, the applicant is directed to column 8, lines 27-40.

26. In addition, the applicant has argued that claims rejected under 35 U.S.C. 103, but not explicitly discussed, are allowable based on the above arguments. Thus, claims disclosing similar limitations to the discussed claims and related dependent claims remain rejected under the same reasoning as presented above.

Conclusion

27. The applicant's amendment necessitated the new grounds of rejection presented in this office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). The applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

Art Unit: 2152

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

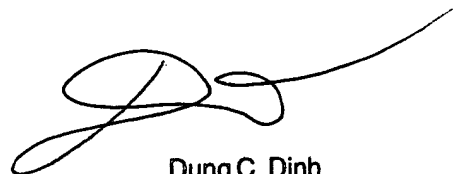
28. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Victor Lesniewski whose telephone number is 571-272-3987. The examiner can normally be reached on Monday through Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Victor Lesniewski
Patent Examiner
Group Art Unit 2152



Dung C. Dinh
Primary Examiner